Performance modeling and evaluation of topologies for low-latency SCI systems

Damian M. Gonzalez, Alan D. George, Matthew C. Chidester

High-performance Computing and Simulation (HCS) Research Laboratory
Department of Electrical and Computer Engineering, University of Florida
P.O. Box 116200, Gainesville, FL 32611-6200

Abstract -- This paper presents an analytical performance characterization and topology comparison from a latency perspective for the Scalable Coherent Interface (SCI). Experimental methods are used to determine constituent latency components and verify the results obtained by these analytical models as close approximations of reality. In contrast with simulative models, analytical SCI models are faster to solve, yielding accurate performance estimates very quickly, and thereby broadening the design space that can be explored. Ultimately, the results obtained here serve to identify optimum topology types for a range of system sizes based on the latency performance of common parallel application demands.

Keywords: Scalable Coherent Interface; Latency; Topology; Analytical modeling; Microbenchmarking

1. Introduction

Modern supercomputing is increasingly characterized by a shift away from the traditional monolithic supercomputing systems toward a new generation of systems using commercial-off-the-shelf (COTS) computers, tightly integrated with high-performance System Area Networks (SANs). Together, these computers and interconnection networks form a distributed-memory multicomputer or *cluster* that offers significantly better price/performance than traditional supercomputers.

A fundamental challenge faced in designing these parallel processing systems is that of interconnect performance. The Scalable Coherent Interface (SCI), ANSI/IEEE Standard 1596-1992 [6] addresses this need by providing a high-performance interconnect specifically designed to support the unique demands of parallel processing systems. SCI offers considerable flexibility in topology choices, all based on the fundamental structure of a ring. However, since a message from one node in a ring must traverse every other node in that ring, this topology becomes inefficient as the number of nodes increases. Multi-dimensional topologies and/or switches are used to minimize the traffic paths and congestion in larger systems.

Before making design decisions between such elaborate topology alternatives, it is first necessary to evaluate the relative performance of available topology choices without incurring the expense of constructing a complete system. Toward this end, this paper presents analytical models for various SCI topologies from a latency perspective, using experimentally-derived parameters as inputs, and validating later against experimental results. These validated models are then used to project tradeoffs between topology choices and their suitability in handling common application demands. Similar topology projections are also performed for a conceptual system featuring enhanced switching performance.

The remainder of the paper is organized as follows. Section 2 provides an overview of related research in this area. Section 3 introduces the fundamentals of SCI communication. Section 4 presents a derivation of analytical models based on these fundamentals. Section 5 provides a description of the experimental testbed, the calculation of

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2001**	2. REPORT TYPE	3. DATES COVERED **00-00-2001 to 00-00-2001**

4. TITLE AND SUBTITLE **Performance modeling and evaluation of topologies for low-latency SCI systems**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Florida,Department of Electrical and Computer Engineering,High-performance Computing and Simulation (HCS) Research Laboratory,Gainesville,FL,32611**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **21**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

experimentally derived input parameters for the models, and a validation of the analytical models against equivalent experimental results. In Section 6, the models are used to predict the performance of topology types that exceed current testbed capabilities. Finally, Section 7 presents conclusions and suggests directions for future research.

2. Related Research

The SCI standard originated out of efforts to develop a high-performance bus that would overcome the inherent serial bottlenecks in traditional memory buses. SCI-related research has since progressed in many different directions, such as the use of SCI in distributed I/O systems [11] and as a SAN interconnect [5].

Significant progress has been made in the use of SCI as a SAN interconnect interfacing with the I/O bus. Hellwagner and Reinefeld [5] present a survey of representative samples of such work, demonstrating results achieved in a variety of related areas, including contributions in the basic definitions, hardware, performance comparisons, implementation experiences, low-level software, higher-level software and management tools.

Simulative models of SCI have been used to investigate issues such as fault tolerance [7] and real-time optimizations [10]. However, simulative modeling often requires several hours to simulate a few seconds of real execution time with any degree of accuracy. An analytical model is orders of magnitude faster to solve, yielding performance estimates very quickly, and thereby broadening the design space that can be explored. Analytical modeling therefore provides a means to project network behavior in the absence of an expensive hardware testbed and without requiring the use of complex, computationally-intensive simulative models.

Analytical modeling of SCI has traditionally focused on cache coherency modeling [1] or queue modeling [9] of SCI components. Relatively little work exists for analytical modeling of SCI developed from an architectural perspective. Such a perspective is necessary to provide insight into scalability and performance as a function of architectural system elements. Such an architecturally motivated analytical model would also offer valuable insight into the suitability of a given system for handling common types of parallel communication behavior.

Horn [3] follows such an architecturally-motivated approach, developing a throughput model for a single ring, and presenting a single chart of results showing the scalability of the SCI ring for different PCI bandwidth capabilities. This model demonstrates scalability issues from a throughput perspective, but does not include a latency study and does not investigate topology types beyond the basic ring. Moreover, no validation of the model used in this study was provided.

Bugge [2] uses knowledge about the underlying hardware, coupled with an understanding of traffic patterns of all-to-all communication to develop an analytical throughput model for all-to-all communication on SCI. He shows the scalability of various multicube topologies ranging from rings to four-dimensional tori. This study makes topology recommendations for varying system sizes, based on a throughput study, but does not include a similar investigation using a latency approach, and does not investigate other types of traffic patterns. This throughput study also lacks a validation exercise.

The simulative study of SCI fault tolerance performed by Sarwar and George [7] presents analytical derivations for average paths taken by SCI request and response packets for one- and two-dimensional topologies, paving the way for extrapolation of topologies to higher degrees. These analytical expressions are used for verification of simulative results, but no validations are made using experimental data.

This paper complements and extends previous work by providing meaningful performance projections of multiple SCI topology types using an architecturally motivated analytical approach. In contrast with existing throughput studies, latency performance is used as a basis for comparison, since for many applications latency has a significant effect on performance. Analytical models are derived and validated against experimental data for traffic patterns that are representative of basic communication in parallel applications. Finally, performance projections are rendered for scalable systems with up to one-thousand nodes in terms of current and emerging component characteristics. The following section provides an overview of SCI communication as background for subsequent development of analytical representations of latency.

3. Overview of SCI

The SCI standard was developed over the course of approximately four years, and involved participation from a wide variety of companies and academic institutions. This standard describes a packet-based protocol using unidirectional links that provides participating nodes with a shared memory view of the system. It specifies transactions for reading and writing to a shared address space, and features a detailed specification of a distributed, directory-based, cache-coherence protocol.

SCI offers many clear advantages for the unique nature of parallel computing demands. Perhaps the most significant of these advantages is its low-latency performance, typically on the order of single-digit microseconds. SCI also offers a link data rate of 4.0 Gb/s in current systems. Yet another advantage in using SCI is that, unlike competing systems, SCI offers support for both the shared-memory and message-passing paradigms.

The analytical latency models developed in this paper rely upon an understanding of the fundamental SCI packet types and the ways in which they interact during a single transaction. A typical transaction consists of two subactions, a *request* subaction and a *response* subaction, as shown in Fig. 1.

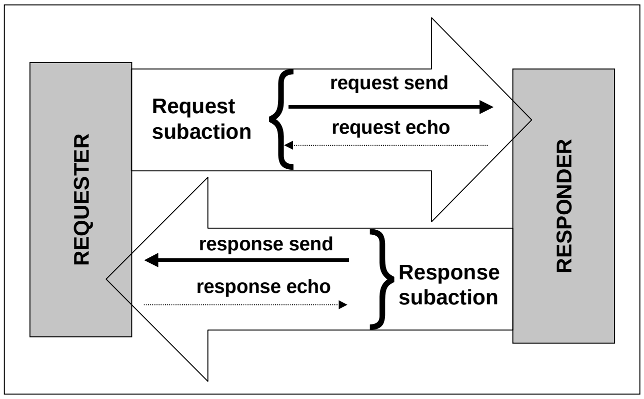


Fig. 1. SCI subactions.

For the request subaction, a request packet (read or write) is sent by a requesting node, destined for a recipient node. The recipient sends an echo back to the requesting node to acknowledge receipt. The recipient simultaneously processes the request and delivers its own response to the network. This packet is received at the original requesting node, and another echo is sent to the recipient to acknowledge receipt of the response packet.

When the source and destination nodes do not reside on the same ring, one or more intermediate *agents* act on behalf of the requester, forwarding the packet along the new ring. In this regard, a node on an SCI torus topology that enacts a change in dimension acts as an agent for that transaction.

In SCI, data is represented in terms of 16-bit (2 byte) *symbols*. All transmissions are conducted based on units of symbols and multiples thereof. Current implementations support both 16-byte (8-symbol) and 64-byte (32-symbol) packet payload sizes. The following section describes the development of analytical representations of SCI transactions using knowledge of these basic packet types and their interaction during an SCI transaction sequence.

4. Analytical Investigation

The topologies considered in this study range from simple rings to multi-dimensional tori as shown in Fig. 2. Subsequent experimentation explores topologies having a maximum of nine nodes and two dimensions, but ultimately analytical models are used to predict the relative performance of systems that far exceed these limits. The analysis of multi-dimensional topologies assumes an equal number of nodes in each dimension. Therefore, for a system with D dimensions and n nodes in each dimension, the *total* number of nodes (i.e. system size) is equal to n^D.

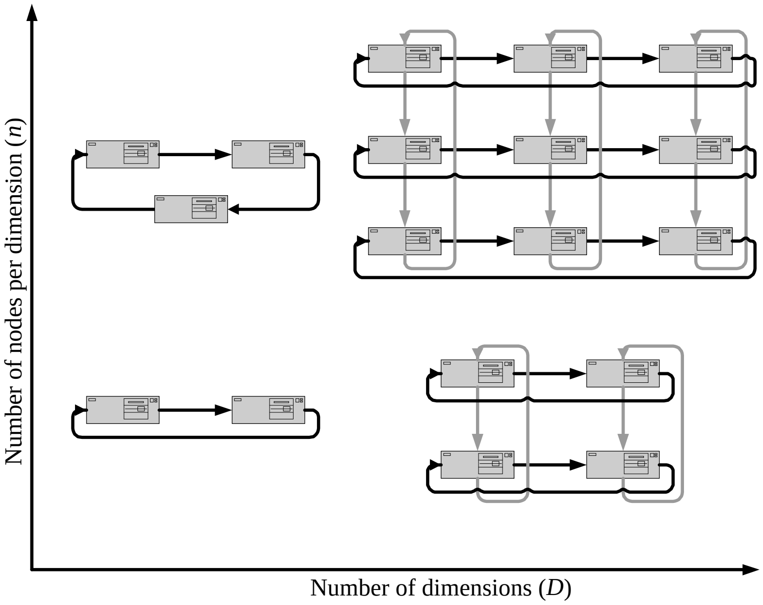


Fig. 2. Topology alternatives.

Considering a point-to-point transaction on a one-dimensional topology, it is assumed that the overhead processing time at the sender is equal to that at the receiver, and these are each represented using the variable o. The variables l_p and l_f represent the propagation latency per hop and the forwarding latency through a node, respectively. The propagation latency is of course dictated by the speed of light through a medium, whereas the forwarding latency is dependent upon the performance of the SCI adapter interface in checking the header for routing purposes and directing the packet onto the output link.

It is important to note that many of these events take place in parallel. For example, for a relatively large packet, the first symbols of the packet may arrive at the recipient before the requester has finished transmission of the complete packet onto the network. This overlap ceases once the time spent by a packet traversing the network is equal to the time spent putting the packet onto the physical links. Using a 16-bit wide path, a 5 ns channel cycle time, and assuming a 40-symbol packet, the time to put this packet onto the link is equal to 200 ns. Using typical values for forwarding and propagation latencies (60 ns and 7 ns respectively), the time spent putting the packet onto the link is matched by hop latencies after traversing only 3 hops. Since any overlapping effect ceases for a relatively small number of hops, the effect of such parallel events does not play a role in subsequent analytical development.

For multi-dimensional tori, there is also a switching latency (l_s) to be considered. This component represents the time taken to switch dimensions from one ring to another ring on a torus. The echo latencies are not considered in this model, since they do not contribute to the critical path of latency.

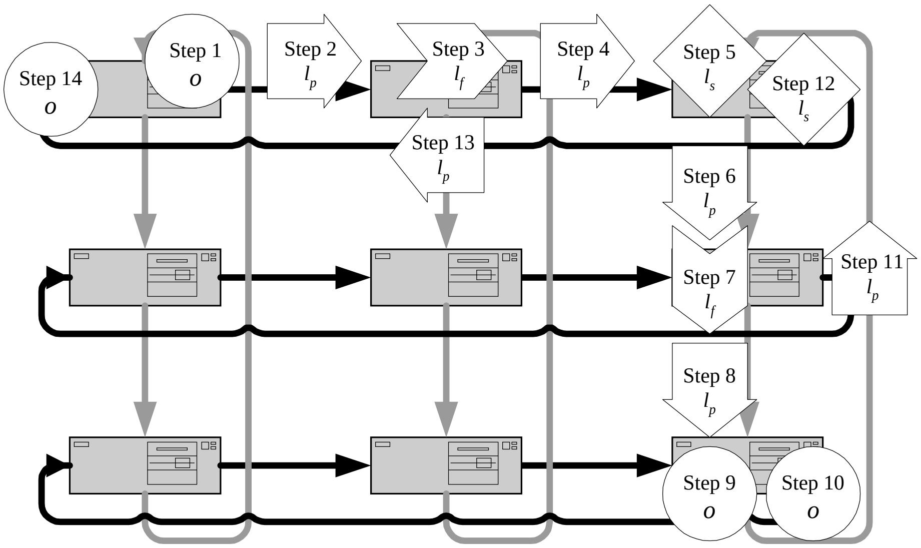


Fig. 3. Latency components for a point-to-point transaction on a 3x3 torus.

Fig. 3 shows how all of these latency components play a role in a complete request and response transaction sequence on a two-dimensional topology. Latency components in the figure are numbered 1 through 14 to identify their ordering in time. Step 1 represents the processing overhead in putting the request packet onto the network. This request packet then incurs forwarding and propagation latencies (Steps 2, 3 and 4) in traversing the horizontal ring. The packet must then switch dimensions (Step 5) and incur forwarding and propagation latencies in traversing the vertical ring (Steps 6, 7 and 8). The request subaction is complete once the recipient incurs the processing overhead for getting the request packet off the network (Step 9).

The response subaction begins with processing overhead for the response packet (Step 10). In traveling back to the original source, this packet incurs a propagation latency along the vertical ring (Step 11), a switching latency (Step 12) and then a propagation latency along the horizontal ring (Step 13). The transaction is complete once the source node incurs the processing overhead for getting the response packet off the network (Step 14).

At this point, it is assumed that the switching, forwarding and propagation latencies will be largely independent of message size, since they only represent the movement of the *head* of a given message. However, the overhead components rely upon the processing of the *entire* message, and are therefore expected to have a dependence upon message size. The validity of these assumptions is investigated through experimental testing in Section 5.

4.1. Point-to-point latency model

Having outlined the latency components that will be considered in this model, it is now necessary to determine the number of times that each of these components will appear for a given point-to-point transaction. Subsequent derivations do not incorporate contention considerations and therefore represent the *unloaded* point-to-point latency.

Consider a point-to-point transaction between two nodes on an SCI network. The overall latency of the transaction is given by:

$$L_{transaction} = L_{request} + L_{response} \tag{1}$$

Using h_k to represent the number of hops from the source to the destination in the k^{th} dimension, the transaction latency components for an SCI ring of n nodes are given by:

$$L_{request} = o + h_1 \times l_p + (h_1 - 1)\times l_f + o \tag{2}$$

$$L_{response} = o + (n - h_1)\times l_p + (n - h_1 - 1)\times l_f + o \tag{3}$$

For a two-dimensional SCI torus with n nodes in each dimension, three cases can occur depending upon the number of hops required in each of the two dimensions. If $h_1 = 0$ or $h_2 = 0$, then the previous equations can be readily applied since the transaction takes place on a single ring. For the third case, where $h_1 \neq 0$ and $h_2 \neq 0$, the request and response latencies are given by:

$$\begin{aligned} L_{request} &= o + h_1 \times l_p + (h_1 - 1)\times l_f + l_s + h_2 \times l_p + (h_2 - 1)\times l_f + o \\ &= 2\times o + [h_1 + h_2]\times l_p + [(h_1 - 1) + (h_2 - 1)]\times l_f + l_s \end{aligned} \tag{4}$$

$$\begin{aligned} L_{response} &= o + (n - h_1)\times l_p + (n - h_1 - 1)\times l_f + l_s + (n - h_2)\times l_p + (n - h_2 - 1)\times l_f + o \\ &= 2\times o + [(n - h_1) + (n - h_2)]\times l_p + [(n - h_1 - 1) + (n - h_2 - 1)]\times l_f + l_s \end{aligned} \tag{5}$$

Using a minimum function to eliminate dimensions with no hop traversals, all three cases are generalized as:

$$\begin{aligned} L_{request} = {} & 2\times o + [h_1 + h_2]\times l_p + [(h_1 - \min(h_1,1)) + (h_2 - \min(h_2,1))]\times l_f \\ & + [\min(h_1,1) + \min(h_2,1) - 1]\times l_s \end{aligned} \tag{6}$$

$$\begin{aligned} L_{response} = {} & 2\times o + [\min(h_1,1)\times(n - h_1) + \min(h_2,1)\times(n - h_2)]\times l_p \\ & + [\min(h_1,1)\times(n - h_1 - 1) + \min(h_2,1)\times(n - h_2 - 1)]\times l_f \\ & + [\min(h_1,1) + \min(h_2,1) - 1]\times l_s \end{aligned} \tag{7}$$

These results are extended for D dimensions as follows:

$$L_{request} = 2\times o + \left[\sum_{i=1}^{D} h_i\right]\times l_p + \left[\sum_{i=1}^{D}\left(h_i - \min\left(h_i, 1\right)\right)\right]\times l_f + \left[\left(\sum_{i=1}^{D}\min\left(h_i, 1\right)\right) - 1\right]\times l_s \tag{8}$$

$$\begin{aligned} L_{response} &= 2\times o + \left[\sum_{i=1}^{D}(\min(h_i,1)\times(n-h_i))\right]\times l_p + \left[\sum_{i=1}^{D}(\min(h_i,1)\times(n-h_i-1))\right]\times l_f \\ &\quad + \left[\left(\sum_{i=1}^{D}\min(h_i,1)\right)-1\right]\times l_s \end{aligned} \tag{9}$$

4.2. Average latency model

Further analysis is now performed to augment the previous point-to-point analysis by characterizing the average distances traveled by request and response packets in a system. The equations below extend the one- and two-dimensional average-distance derivations of Sarwar and George [7] into a general form for *D* dimensions.

First, consider a single ring, and assume that there is a uniformly random distribution of destination nodes for all packets. To arrive at the average number of links traversed in a ring, a scenario having a fixed source and variable destinations is considered. The *total* distance traveled for *all* possible source/destination pairs is determined, and then divided by the number of destinations to determine the average distance traveled.

The variable h_1 is used to represent the number of hops in the single dimension for a given source/destination pair. For a request that has traveled h_1 hops, the response will travel $n - h_1$ hops around the remainder of the ring. Therefore, the average number of hops for request and response packets in a ring is represented as follows:

$$Average\ \ request\ \ distance = \frac{\sum_{h_1=0}^{n\text{-}1} h_1}{n-1} = \frac{n}{2} \tag{10}$$

$$Average\ \ response\ \ distance = \frac{\sum_{h_1=0}^{n\text{-}1}\left(n-h_1\right)}{n-1} = \frac{n}{2} \tag{11}$$

Similarly, for a two-dimensional system, using h_2 to represent the number of hops in the second dimension, the derivation for average number of hops is as follows:

$$Average\ \ request\ \ distance = \frac{\sum_{h_1=0}^{n\text{-}1}\left(\sum_{h_2=0}^{n\text{-}1}\left(h_1+h_2\right)\right)}{n^2-1} = \frac{n^2\times\left(n-1\right)}{n^2-1} \tag{12}$$

$$Average\ \ response\ \ distance = \frac{\sum_{h_1=0}^{n\text{-}1}\left(\sum_{h_2=0}^{n\text{-}1}\left(\left(n-h_1\right)+\left(n-h_2\right)\right)\right)}{n^2-1} = \frac{n^2\times\left(n-1\right)}{n^2-1} \tag{13}$$

Based on the results of similar derivations for three- and four-dimensional systems, a general expression is derived for the average number of hops as a function of *D* dimensions:

$$Average\ \ request\ \ distance = Average\ \ response\ \ distance = \frac{D}{2}\times\frac{n^D\times\left(n-1\right)}{n^D-1} \tag{14}$$

Relating this result to the original latency model, the average number of hops that these equations describe can therefore be used to represent the number of propagation latencies for a given point-to-point transaction.

As for switching latencies, it can be shown that the average number of dimension switches for a transaction in a torus of *D* dimensions is accurately represented as follows:

$$\text{Average number of dimension switches} = \frac{\sum_{i=1}^{D}\left[(i-1)\times\binom{D}{i}\times(n-1)^{i}\right]}{n^{D}-1} \tag{15}$$

For a single ring, the number of forwarding latencies is always one less than the number of propagation latencies. However, when considering a transaction on a multi-dimensional topology, the sum of the number of forwarding and switching latencies is one less than the number of propagation latencies. Using the preceding analysis for the average number of switching latencies (Eq. 15) and propagation latencies (Eq. 14), the number of forwarding latencies can be determined as follows:

$$\begin{aligned} &\mathit{Average\ number\ of\ forwarding\ delays} + \frac{\sum_{i=1}^{D}\left[(i-1)\times\binom{D}{i}\times(n-1)^{i}\right]}{n^{D}-1} = \frac{D}{2}\times\frac{n^{D}\times(n-1)}{n^{D}-1}-1 \\ &\mathit{Average\ number\ of\ forwarding\ delays} = \frac{\left(\frac{D}{2}\times n^{D}\times(n-1)\right)-\sum_{i=1}^{D}\left[(i-1)\times\binom{D}{i}\times(n-1)^{i}\right]}{n^{D}-1}-1 \end{aligned} \tag{16}$$

Therefore, Eq. 17 represents the average latency of a request or response packet in a *D*-dimensional topology.

$$\begin{aligned} L_{request} = L_{response} = 2\times o + \left(\frac{D}{2}\times\frac{n^{D}\times(n-1)}{n^{D}-1}\right)\times l_{p} + \left(\frac{\left(\frac{D}{2}\times n^{D}\times(n-1)\right)-\sum_{i=1}^{D}\left[(i-1)\times\binom{D}{i}\times(n-1)^{i}\right]}{n^{D}-1}-1\right)\times l_{f} \\ + \left(\frac{\sum_{i=1}^{D}\left[(i-1)\times\binom{D}{i}\times(n-1)^{i}\right]}{n^{D}-1}\right)\times l_{s} \end{aligned} \tag{17}$$

In the following section, experimental methods are used to determine the values for overhead, switching, forwarding and propagation latency to be used as inputs for these analytical models.

5. Experimental Investigation

Experimental results described in this paper were obtained using an SCI testbed consisting of nine PCs, each having one 400 MHz Intel Pentium-II processor and 128 MB of PC100 SDRAM. These PCs each contained 32-bit

PCI bus interfaces operating at 33 MHz, and were connected using Dolphin/Scali Wulfkit [8] adapters having a link data rate of 4.0 Gb/s. Experimental performance measurements were obtained by configuring this system in a variety of ring and torus topologies and were taken from an otherwise unloaded system.

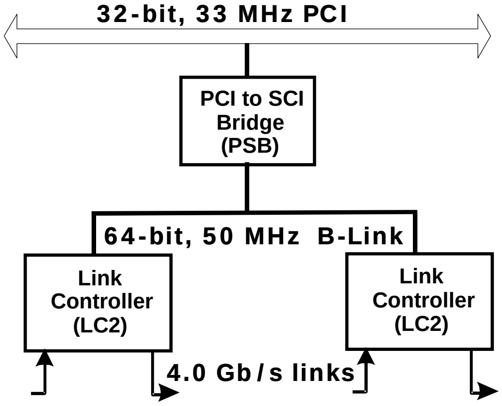


Fig. 4. Architectural components of a Wulfkit SCI NIC.

Fig. 4 shows a block diagram of the main components of the NIC for a single node in any given topology. A request originating at this node will enter the NIC through the PCI bus, at which point the PCI to SCI Bridge (PSB in Fig. 4) transfers this data from PCI to the internal B-link bus. The request send packet then traverses the B-link, and enters the SCI network fabric through one of the Link Controllers (LC2 in Fig. 4). Together with the software overhead for processing the message on the host, these steps collectively constitute the sender overhead (o).

Packets entering the NIC from one of the SCI links will first encounter an LC2, and that controller will check the header for routing purposes. Three cases can occur based on the information contained in the packet header. In the first case, the header address could correspond with the address of the node, and in this case the packet would traverse all intermediate components and enter the PCI bus for further processing by the host. Together with the associated software overhead, these steps constitute the receiver overhead component in the analytical model (o).

Another possibility is that the packet is destined for a node that resides on another SCI ring for which this node can serve as an agent. In such a case, the packet is sent across the internal B-link bus, and enters the second ring through another LC2. These components correspond with the switching delay (l_s) in the analytical study.

In the third possible scenario, the incoming packet is addressed to a different node residing on the same ring. In this case, the packet is routed back out the same LC2, without traversing any other NIC components. These steps correspond with the forwarding delay (l_f) in the analytical study.

The Red Hat Linux (kernel 2.2.15) operating system was used on all machines in the testbed, and each machine contains 100 Mb/s Ethernet control networks. The Scali Software Platform 2.0.0 provides the drivers, low-level API (SCI USRAPI), and MPI implementation (ScaMPI [4]) to support experimental testing.

Although MPI results feature additional software overhead, they provide an idea of the performance offered at the application level. Fig. 5 clearly demonstrates the performance penalty paid for the simplicity and portability of MPI, providing a comparison of the Scali USRAPI and ScaMPI on a two-node ring using a one-way latency test (see Fig. 6). The API results are consistently lower, achieving a minimum latency of 1.9 µs for a one-byte message, while the minimum latency achieved using ScaMPI is 6.4 µs for a similarly sized message.

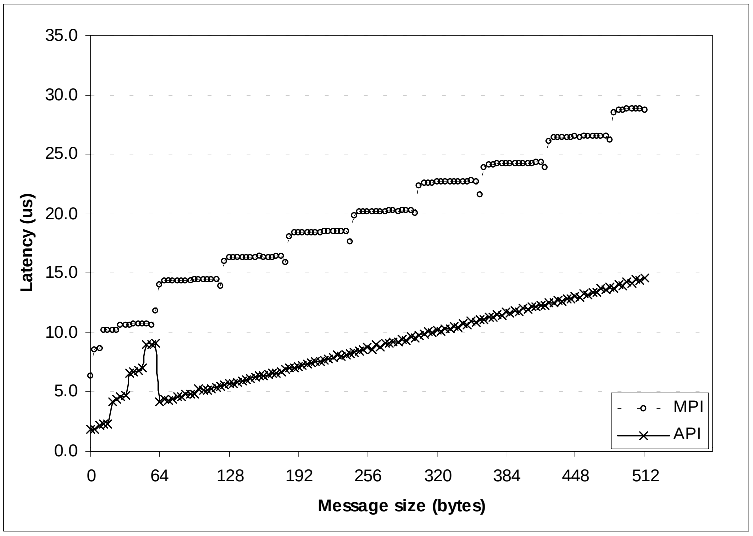


Fig. 5. Comparison of MPI and API latency on a two-node ring.

The results shown in Fig. 5 demonstrate another important point. The shape of the curves for both API and MPI suggests that the overall trend in behavior is dominated by the performance of SCI transactions having a 64-byte payload size. Transactions with 16-byte packet payloads occur only for message sizes less than 64 bytes. Subsequent analysis focuses on the behavior of SCI transactions having a 64-byte packet payload size.

Since the experimental latency results are on the order of microseconds, transient system effects (e.g. context switching, interrupts, cache misses, UNIX watchdog, etc.) and timer resolution issues can negatively affect the results even on an unloaded system. The results shown here represent the *minimum* value obtained from a series of experiments of sufficient duration as to provide less than five percent variation from run to run.

5.1 Benchmark design

For both API- and MPI-based testing, many design alternatives are available. A suite of benchmarks was developed as part of this work to support both MPI (`mpibench`) and API (`scibench`) benchmarking. Both one-way (OW) and ping-pong (PP) latency tests were used. Figs. 6 and 7 explain these strategies using relevant pseudo-code. The OW test in Fig. 6 uses one sender and one receiver to measure the latency of uni-directional streaming message transfers. The PP test in Fig. 7 alternates the roles of sender and receiver for each message transferred, and the PP latency is computed as half the round-trip time.

`Mpibench` performs both tests, and the standardization of the interface allows it to be easily ported for use in other high-performance networking systems. `Scibench` also performs both tests, using API calls to establish the sharing and mapping of remote memory.

The API-based benchmarks use local reads (for polling a memory location for changes) and remote writes, since remote polling would incur a significant and unnecessary performance penalty. Writes of large messages are performed by assigning a remote target to `memcpy()` function calls. However, when transferring messages of 8 bytes and smaller, direct assignments of atomic data types (`char, short, int, long long int`) are used to avoid the overhead of the `memcpy()` function call.

```
SERVER:

start timing
  for (index=0; index < reps; index++)
  {
     send message of predetermined size;
  }
  receive acknowledgement of final message;
end timing

'One-way latency'=(end-start)/reps;
```

```
CLIENT:


  for (index=0; index < reps; index++)
  {
     receive message of predetermined size;
  }
  send acknowledgement of final message;
```

Fig. 6. One-way (OW) testing scheme.

```
SERVER:

start timing
  for (index=0; index < reps; index++)
  {
     send message of predetermined size;
     receive message of predetermined size;
  }
end timing

'Ping-pong latency'=((end-start)/reps)/2;
```

```
CLIENT:


  for (index=0; index < reps; index++)
  {
     receive message of predetermined size;
     send message of predetermined size;
  }
```

Fig. 7. Ping-pong (PP) testing scheme.

To investigate the components of latency, multiple experiments were performed on different topology types and the results compared to determine the transaction latency components. The next subsection focuses upon experiments with a ring topology, and is followed by experiments with a torus topology.

5.2 Ring experiments

In the first series of experiments, several configurations based on a ring topology were used to conduct PP latency testing. PP tests were used since the analytical derivations are based on the behavior of a single message, and the compound effect of multiple serial messages in an OW test would necessarily feature a potentially misleading pipeline effect.

Fig. 9 analyzes the execution of a PP test on SCI, with constituent steps numbered 1 through 8 to identify the ordering of ping and pong transaction components. Steps 1 through 4 describe the four components (see Fig. 1) of a single ping transaction. Steps 5 through 8 represent the subaction components for the corresponding pong transaction.

The ping-pong latency is calculated to be half the round-trip time, and is shown to be equivalent to the timing of a single ping request (step 1 in Fig. 8). Therefore, the analytical representation of a ping request ($L_{request}$) is used in subsequent analysis to represent experimental PP results.

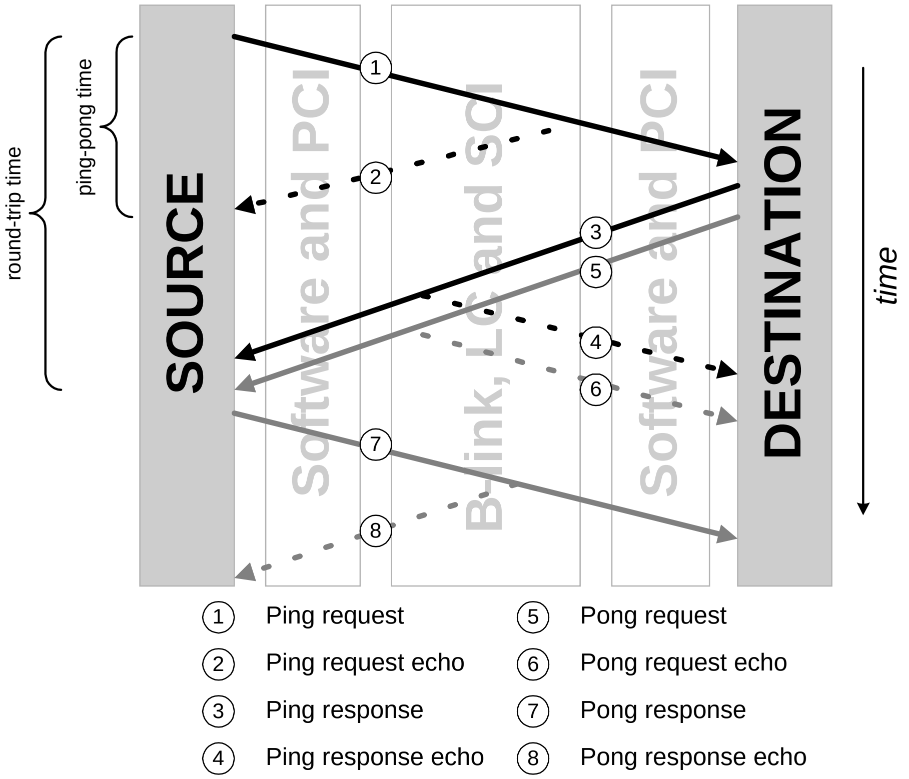
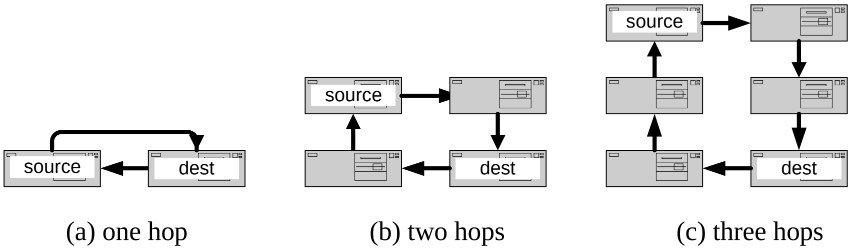


Fig. 8. Analysis of PP testing.

The propagation latency (l_p) was determined theoretically, by considering the fact that signals propagate through a conductor at approximately half the speed of light. Using a value of 299,792.5 km/s for the speed of light, and assuming cables one meter in length, the propagation latency was determined to be 7 ns. Since the propagation latency represents the latency for the head of a message passing through a conductor, it is therefore independent of message size.



(a) one hop (b) two hops (c) three hops

Fig. 9. Ring test configurations.

Client and server nodes were chosen such that there is a symmetrical path between source and destination. Fig. 9 demonstrates three such testing configurations in which tests are identified based on the number of hops traversed by a ping request traveling from the source to destination. These path scenarios were selected so that the *ping* and *pong* messages would each traverse the same number of hops. Such symmetrical paths allow the PP result to be characterized by an integer number of hops, facilitating a direct association of an experimental PP result with the corresponding analytical representation of a ping request. Once these experimental results were obtained, their differences were then used to determine latency components.

The first experiment is designed to determine the value of overhead, and involves the one-hop test shown in Fig. 9a. PP latency was measured over a range of message sizes, and is represented analytically as follows:

$$PP_{one\ hop} = 2 \times o + l_p \tag{18}$$

Since l_p has already been determined (7 ns), the overhead component is the only unknown in Eq. 18. This overhead was computed algebraically for a range of message sizes, and the results of this computation are discussed further in the next subsection.

The next series of ring experiments applied the difference between PP latencies for the one-hop test and similar results obtained from a four-hop test. The difference between these is derived analytically as follows:

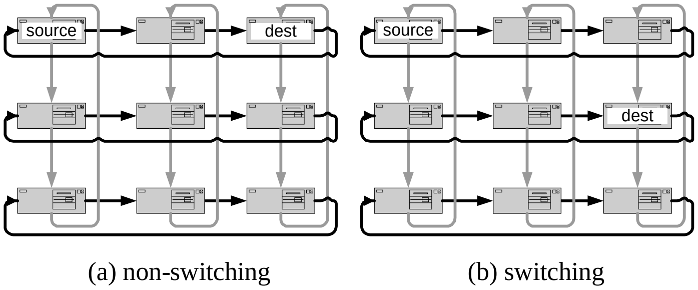
$$\begin{aligned} PP_{four\ hops} &= 2 \times o + 3 \times l_f + 4 \times l_p \\ PP_{one\ hop} &= 2 \times o + l_p \\ PP_{four\ hops} - PP_{one\ hop} &= 3 \times l_f + 3 \times l_p \end{aligned} \tag{19}$$

Using the value previously obtained for propagation latency, the forwarding latency is the only unknown in Eq. 19. As such, the value for forwarding latency was computed algebraically for a range of message sizes. The results from this derivation are also discussed further in the next subsection.

The only remaining unknown is the switching latency component, which occurs when a message switches dimensions from one ring to another. This component is determined using a series of torus experiments.

5.3 Torus experiments

The switching latency is determined using PP benchmarking for the torus-based testing scenarios shown in Fig. 10. Fig. 10a illustrates the first test configuration, which involves movement in a single dimension. The second configuration, shown in Fig. 10b, introduces the switching latency element.



(a) non-switching (b) switching

Fig. 10. Torus test configurations.

Once the latency experiments were performed for a range of message sizes on each of these two configurations, the difference between the two sets of results was determined algebraically. Although the topology in Fig. 10a is no longer perfectly symmetrical for ping and pong paths, the following provides a close approximation of the algebraic difference between torus experiments:

$$
\begin{aligned}
PP_{non\text{-}switching} &\cong 2 \times o + 1 \times l_f + 2 \times l_p \\
PP_{switching} &= 2 \times o + 1 \times l_f + 3 \times l_p + l_s \\
PP_{switching} - PP_{non\text{-}switching} &\cong l_s + l_p \qquad (20)
\end{aligned}
$$

As before, Eq. 20 is used along with the value for propagation latency to algebraically determine the switching latency for a range of message sizes.

Using Eqs. 18, 19 and 20 along with the predetermined value for propagation latency, Fig. 11 shows a comparison of all components for a range of message sizes. This figure demonstrates the clear differences between components in terms of their relationship with message size. The switching, forwarding and propagation components are shown to be relatively independent of message size, whereas the overhead component is significantly dependent upon message size.

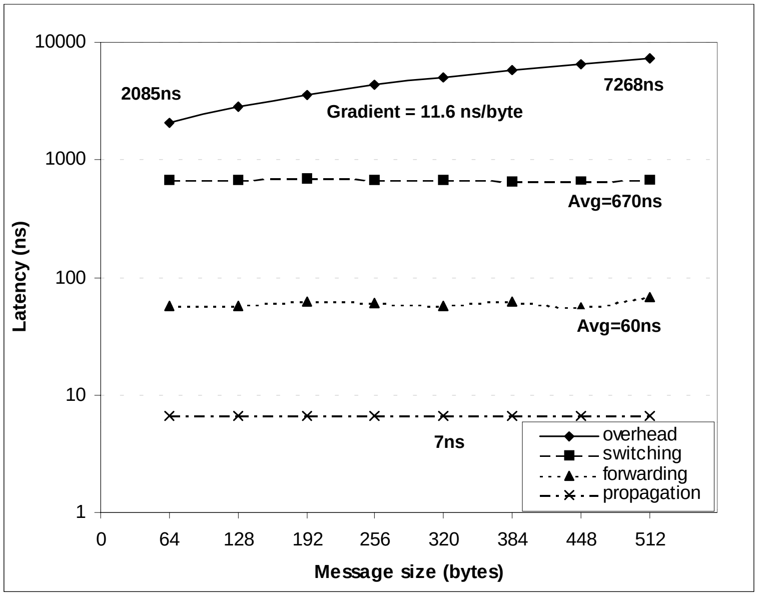


Fig. 11. Comparison of latency components.

Table 1
Estimates of experimental latency components

Latency component	Estimate
Propagation latency (l_p)	7 ns
Forwarding latency (l_f)	60 ns
Switching latency (l_s)	670 ns
Overhead (o)	$2085 + 11.6 \times (m - 64)$ ns

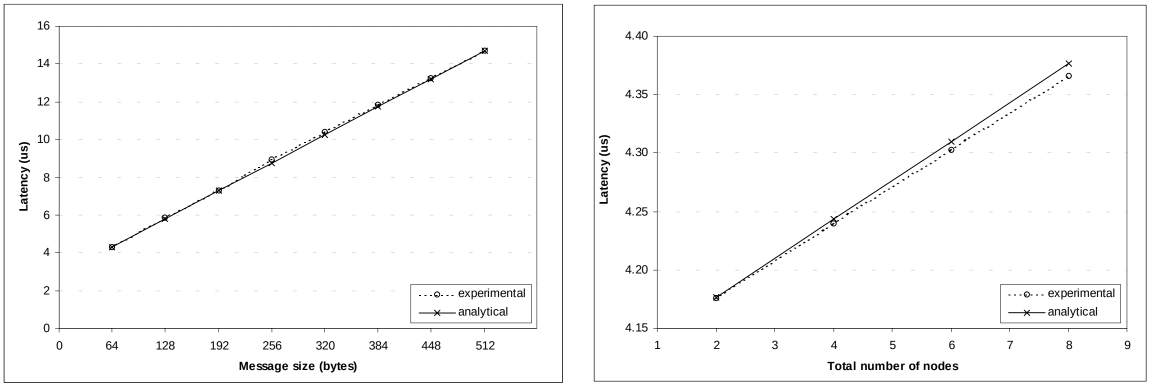
Table 1 provides a summary of the estimates made for each component of the analytical model for a message of *m* bytes. Propagation, forwarding and switching components are assumed constant, whereas the overhead component is represented using a linear equation.

5.4 Validation

Using these estimates as inputs to the analytical models, a validation exercise is performed to confirm the models as worthy representations of reality. The first validation exercise investigates the accuracy of the model as a function of message size, and involves the symmetrical three-hop ring test shown in Fig. 9c. This test is chosen because one- and four-hop tests were used previously to experimentally determine the inputs. The analytical PP latency for the three-hop test is given by the following equation:

$$PP_{three\ hops} = 2 \times o + 2 \times l_f + 3 \times l_p \qquad (21)$$

Fig. 12a shows the results of this validation, and demonstrates how closely the analytical estimates match experimental results.



(a) latency vs. message size (6-node ring)

(b) latency vs. ring size (64-byte message)

Fig. 12. Validation of analytical model.

The second validation exercise investigates the accuracy of the model as a function of the number of nodes and uses a 64-byte message size on one-, two-, three-, and four-hop tests. The analytical PP latencies for these rings are given by the following equations:

$$PP_{one\ hop} = 2 \times o + 0 \times l_f + 1 \times l_p \quad (22)$$

$$PP_{two\ hops} = 2 \times o + 1 \times l_f + 2 \times l_p \quad (23)$$

$$PP_{three\ hops} = 2 \times o + 2 \times l_f + 3 \times l_p \quad (24)$$

$$PP_{four\ hops} = 2 \times o + 3 \times l_f + 4 \times l_p \quad (25)$$

Fig. 12b shows the results of this validation, demonstrating the accuracy of the analytical estimates. Although a slight deviation is observed between analytical and experimental results, a linear extrapolation of this deviation for systems sizes up to one-thousand nodes shows that the error never exceeds five percent within this range.

Having now derived and validated the analytical models, the next section uses these models to project the behavior of larger systems. Performance projections are considered for large-scale systems based on both existing and emerging forms of the technology and the architectures motivated by them.

6. Analytical projections

To ascertain the relative latency performance of different topology types, the analytical models were used to investigate topologies that range from one to four dimensions, with a maximum system size of up to one-thousand nodes. The models were first given input parameters derived directly from the experimental analysis. Based on the results of these analytical projections, a conceptual system featuring enhanced parameters is studied.

Two types of applications are considered in determining topology tradeoffs. The first type is average latency, based on the equations derived in Section 4.2. This application provides a useful performance metric since it represents the performance that is achieved in a typical point-to-point transaction on a given topology.

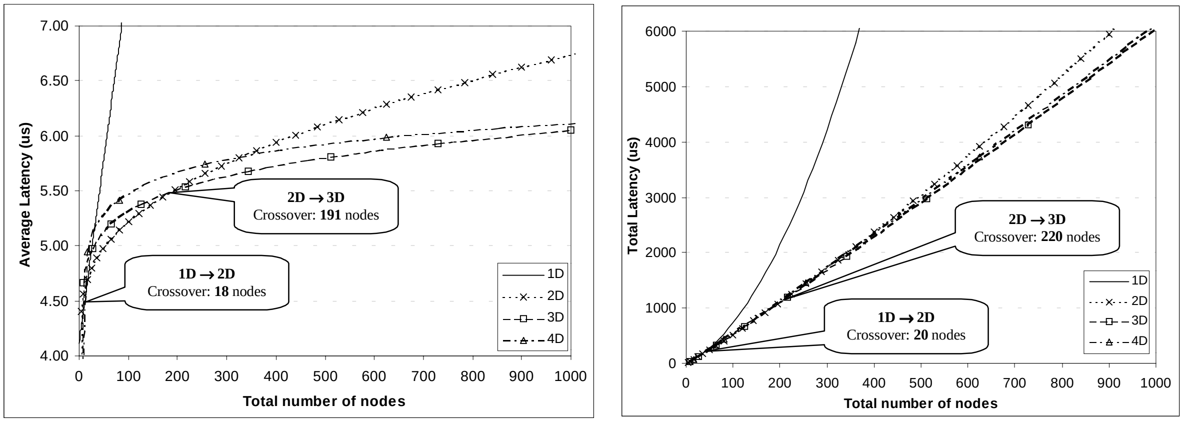
The second application type used for comparison is an unoptimized broadcast operation, carried out using a series of unicast messages. For a given topology, having a fixed source, the complete set of destinations is determined, and the point-to-point latency of each of these transactions is calculated using the latency equations derived in Section 4.1. As before, each point-to-point transaction is assumed equivalent to the latency of a ping request, based on the analysis in Fig. 8. The sum of these transactions is determined, and is used as a basis for inter-topology comparison. This *one-to-all multi-unicast* operation is also a useful metric for comparison, since such an approach for collective communication operations is common in parallel applications and systems.

6.1 Current system

To investigate the relative latency performance of different topology alternatives using current hardware, the performance of average latency and one-to-all multi-unicast applications was derived analytically using data obtained directly from the experimental testing. The results obtained are shown in Fig. 13. The *crossover* points identify the system sizes after which an incremental increase in dimensionality offers superior latency performance.

Since this study was conducted using topologies having equal numbers of nodes in each dimension, the ring is the only featured topology that can offer every system size within the range of interest. An extrapolation of the higher-dimensional topologies was used to fill in the gaps and identify the exact crossover points at which certain topology types surpass the performance of others. For this reason, crossover points do not necessarily align with equi-dimensional topology types, but they still provide a useful basis for inter-topology comparisons.

The average latency application, shown in Fig. 13a, demonstrates clear scalability differences between topologies, with a one-dimensional topology offering the best latency performance for systems having fewer than 18 nodes. Of course, with a simple ring, all nodes must share a common communication path, thus limiting scalability. Beyond 18 nodes, the two-dimensional topology is able to distribute the traffic paths to a sufficient extent to outweigh the large dimension-switching penalty paid as the number of dimensions increases.



(a) average latency

(b) one-to-all multi-unicast

Fig. 13. Inter-topology comparison of current system.

The two-dimensional topology continues to lead latency performance up to 191 nodes, at which point the additional path savings achieved using a three-dimensional topology now outweighs the added switching latency for this higher-dimensional topology. The three-dimensional topology continues to lead latency performance up to 1000 nodes and beyond.

The situation for the one-to-all multi-unicast application, shown in Fig. 13b, is quite different. The savings achieved in going from one to two dimensions is pronounced, but for higher dimensions, the relative latency performance does not vary much within the range of interest. For this application, one-dimensional topologies lead latency performance for system sizes smaller than 20 nodes, at which point the path savings of the two-dimensional topology enables it to provide the best latency performance up to 220 nodes. The three-dimensional topology then offers the best latency performance up to 1000 nodes and beyond.

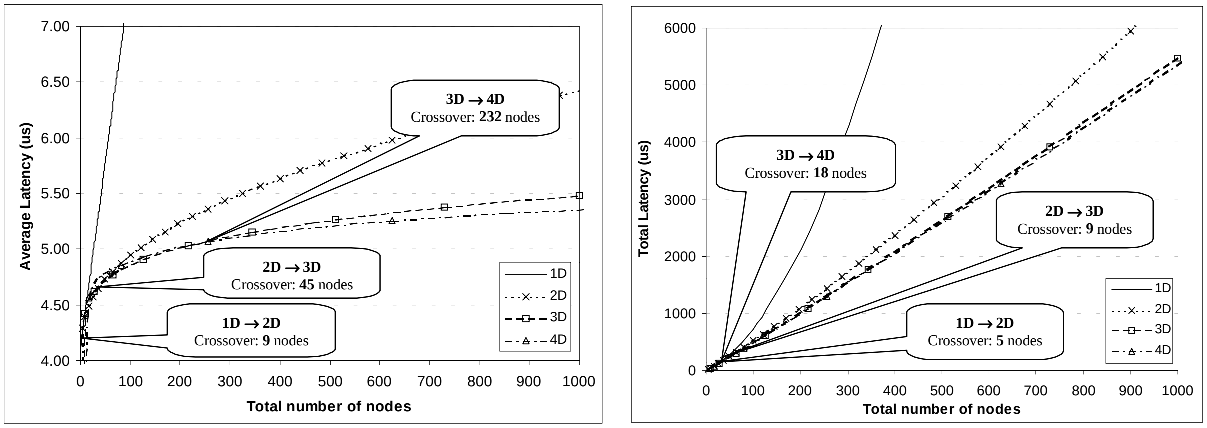
These results demonstrate that one- and two-dimensional topologies dominate latency performance for small and medium system sizes. Crossover points depend primarily upon the relative magnitude of switching and forwarding delays. Although higher-dimensional topologies offer significant path savings for point-to-point traffic, the associated switching penalty makes these topologies impractical for medium-sized systems.

As a means of comparison, these results somewhat mirror those achieved by Bugge [2], who performed similar comparisons of multi-dimensional torus topologies based on a throughput study. Though the crossover points differ, similar conclusions are drawn, with higher-dimensional topologies becoming practical for very large system sizes.

6.2 Enhanced system

Advances in semiconductor manufacturing techniques have been able to sustain a breathtaking pace of improvement in related technologies. As such, it is reasonable to expect that systems will be available in the near future that significantly outperform current hardware. To investigate the relative latency performance of different SCI topology alternatives using such enhanced hardware, the analytical models were fed data that artificially enhanced system performance.

The component calculations in Fig. 11 demonstrate the order of magnitude difference between switching and forwarding latencies (670 ns and 60 ns respectively). The topology comparisons in Fig. 13 demonstrate that this large difference limits the practicality of higher-dimensional topologies for medium-sized systems. An improvement in the switching latency parameter should therefore produce promising results. To examine latency performance of hardware having an enhanced switching latency, the original value (670 ns) is halved (335 ns) to explore the potential effect on performance. Fig. 14 shows the results achieved after making this change.



(a) average latency

(b) one-to-all multi-unicast

Fig. 14. Inter-topology comparison of enhanced system.

The average latency application, shown in Fig. 14a, once again demonstrates clear differences between topologies. The one-dimensional topology is now outperformed by the two-dimensional topology for a system size above 9 nodes. The two-dimensional topology leads latency performance until 45 nodes, at which point the three-dimensional topology leads latency performance up to 232 nodes. The four-dimensional topology becomes a consideration from this point up to 1000 nodes and beyond.

The one-to-all multi-unicast application performance, as shown in Fig. 14b, reflects similar trends to those in the previous configuration. The savings achieved in going from one to two dimensions is again more pronounced than subsequent dimension increases, but there is a clear downward shift overall as the crossover points all occur for smaller system sizes. One-dimensional topologies are quickly outperformed by the two-dimensional topology (5 nodes), which then leads latency performance up to only 9 nodes, at which point the three-dimensional topology leads latency performance up to only 18 nodes. The four-dimensional topology marginally leads in latency performance for the remaining range of system sizes.

Although the crossover points achieved for multi-unicast on enhanced hardware are significantly smaller than those achieved using current hardware, this downward shift is not as significant as the average latency case since the *best* latency performance for a given system size does not improve as significantly in the multi-unicast comparison (Fig. 14b) as it does in the average latency comparison (Fig. 14a). Table 2 summarizes the crossover points for average latency and one-to-all multi-unicast applications using both the current system and the enhanced system. This table shows the system sizes at which the path savings of each dimension increase outweighs the associated switching penalty, facilitating superior latency performance for the higher-dimensional topology in each case.

Table 2
Summary of crossover points (in nodes)

Crossover point	**Current system**		**Enhanced system**	
	Average latency	Multi-unicast	Average latency	Multi-unicast
1D → 2D	18	20	9	5
2D → 3D	191	220	45	9
3D → 4D	1831	2050	232	18

The results indicate that an improvement in the switching latency (e.g. achieved perhaps through the use of a wider or faster internal B-link bus) would enable higher-dimensional topologies to become beneficial for smaller system sizes. Such a design enhancement would provide improved latency performance, but this may or may not justify the added complexity of a higher-dimensional topology.

The average latency data suggests that the enhancement may be warranted, since the *best* latency performance for medium-sized systems is seen to improve, although only by a modest amount (e.g. approx. 5% improvement for a system size of 100 nodes). However, for the one-to-all multi-unicast application, the enhanced system offers no real improvement for medium system sizes. The enhanced hardware only offers an improvement in multi-unicast performance for large system sizes (e.g. approx. 10% improvement for a system size of 1000 nodes).

7. Conclusions

This paper introduces an analytical characterization of SCI network performance and topology comparison from a latency perspective, using architectural issues to inspire the characterization. Analytical models were developed for point-to-point and average latency of various topology types, and a validation exercise demonstrated that these models closely match equivalent experimental results. Based on these models, this work helps determine architectural sources of latency for various systems and provides a straightforward means to project network behavior in the absence of an expensive hardware testbed and without requiring the use of computationally-intensive simulative models. These results should serve as a valuable means for system designers to obtain accurate predictions of the relative performance of topology alternatives.

Using system parameters derived from experimental testing, topology differences for a range of system sizes are found to be a result of the large difference between forwarding latencies and switching latencies. Analytical projections demonstrate the tradeoffs between path savings on higher-dimensional topologies versus the large switching penalty paid when increasing the number of dimensions.

One-dimensional topologies offer superior latency performance for small numbers of nodes, but are soon outperformed by two-dimensional topologies due to the inherent lack of scalability of the basic SCI ring. Using current hardware, the two-dimensional topology continues to lead latency performance for medium system sizes (ranging approximately from 20 nodes to 200 nodes). For larger system sizes, the three-dimensional topology provides the best latency performance for the remainder of the range of interest. When using an enhanced system with a smaller switching latency, higher-dimensional topologies become favorable for medium-sized systems, but the improvement in *best* latency performance for such system sizes is minimal.

In terms of future directions for this research, although the current models provide an accurate approximation of the experimental data, they can be further elaborated to include finer-grained representations of constituent network events. These improvements could involve investigating more subtle phenomena (e.g. contention issues) thereby enhancing the fidelity and usefulness of the models. In terms of experimental analysis, further testbed work could involve more elaborate topology alternatives including larger numbers of nodes, bi-directional rings, faster network/host interfaces, and switch-inclusive studies.

In addition, while the average latency and one-to-all multi-unicast applications provide a practical comparison of topology types, opportunity exists for the study of more types of traffic patterns than the ones investigated here. Some examples of such application patterns include all-to-all, nearest-neighbor, unbalanced communication and tree-based multicasting. Ultimately, such enhancements can be used to predict the behavior of more complex parallel applications, and map these applications to the topology types that best serve their needs.

Acknowledgements

We gratefully acknowledge the support of our sponsors at the U.S. Department of Defense, Dolphin Interconnect Solutions Inc., and Scali Computer AS. We also thank Håkon Bugge at Scali and Kyu-Sang Park at the University of Florida for their timely and valuable assistance throughout the development of this work.

References

[1] A. Bennett, A. Field, P. Harrison, Modeling and Validation of Shared Memory Coherency Protocols, Performance Evaluation 28 (1996) 541-562.

[2] H. Bugge, Affordable Scalability using Multicubes, in: H. Hellwagner, A. Reinefeld (Eds.), SCI: Scalable Coherent Interface, LNCS State-of-the-Art Survey (Springer, Berlin, 1999) 167-174.

[3] G. Horn, Scalability of SCI Ringlets, in: H. Hellwagner, A. Reinefeld (Eds.), SCI: Scalable Coherent Interface, LNCS State-of-the-Art Survey, (Springer, Berlin, 1999) 151-165.

[4] L. Huse, K. Omang, H. Bugge, H. Ry, A. Haugsdal, E. Rustad, ScaMPI - Design and Implementation, in: H. Hellwagner, A. Reinefeld (Eds.), SCI: Scalable Coherent Interface, LNCS State-of-the-Art Survey (Springer, Berlin, 1999) 249-261.

[5] H. Hellwagner, A. Reinefeld, SCI: Scalable Coherent Interface, LNCS State-of-the-Art Survey (Springer, Berlin, 1999).

[6] IEEE, SCI: Scalable Coherent Interface, IEEE Approved Standard 1596-1992, 1992.

[7] M. Sarwar, A. George, Simulative Performance Analysis of Distributed Switching Fabrics for SCI-based Systems, Microprocessors and Microsystems 24 (1) (2000) 1-11.

[8] Scali Computer AS, Scali System Guide Version 2.0, White Paper, Scali Computer AS, 2000.

[9] S. Scott, J. Goodman, M. Vernon, Performance of the SCI Ring, in: Proceedings of the 19th Annual International Symposium on Computer Architecture, Gold Coast, Australia, May 1992, pp. 403-414.

[10] W. Windham, C. Hudgins, J. Schroeder, M. Vertal, An Animated Graphical Simulator for the IEEE 1596 Scalable Coherent Interface with Real-Time Extensions, Computing for Engineers 12 (1997) 8-13.

[11] B. Wu, The Applications of the Scalable Coherent Interface in Large Data Acquisition Systems in High Energy Physics, Dr.Scient. thesis, Department of Informatics, University of Oslo, Norway, 1996.